

java assignment 2

Name - Aditya Mishra

Roll no - 23/SCA/BCA(AI&ML)/002

Class & sec - 4th-c

```
import javax.swing.*;
```

```
import java.awt.*;
```

```
import java.awt.event.*;
```

```
import java.io.*;
```

```
import java.util.*;
```

```
class Book implements Serializable {
```

```
    private String title;
```

```
    private String author;
```

```
    private boolean isIssued;
```

```
    public Book(String title, String author) {
```

```
        this.title = title;
```

```
        this.author = author;
```

```
        this.isIssued = false;
```

```
    }
```

```
    public String getTitle() { return title; }
```

```
    public String getAuthor() { return author; }
```

```
    public boolean isIssued() { return isIssued; }
```

```
public void toggleIssue() { isIssued = !isIssued; }

public String toString() {
    return title + " by " + author + " [" + (isIssued ? "Issued" : "Available") + "];"
}
}
```

```
public class LibraryManagementSystemGUI extends JFrame {

    private static final String FILE_NAME = "books.dat";

    private ArrayList<Book> books;

    private JTextArea displayArea;

    private JTextField titleField, authorField;

    public LibraryManagementSystemGUI() {

        setTitle("Library Management System");

        setSize(500, 400);

        setDefaultCloseOperation(EXIT_ON_CLOSE);

        setLayout(new BorderLayout());

        books = loadBooks();

        // Set up the input panel (for adding books)

        JPanel inputPanel = new JPanel();

        inputPanel.setLayout(new FlowLayout());

        inputPanel.add(new JLabel("Title:"));

        titleField = new JTextField(15);

        inputPanel.add(titleField);
```

```
inputPanel.add(new JLabel("Author:"));

authorField = new JTextField(15);

inputPanel.add(authorField);

JButton addBtn = new JButton("Add Book");

inputPanel.add(addBtn);


// Buttons for actions

JPanel buttonPanel = new JPanel(new GridLayout(2, 3, 10, 10));

JButton removeBtn = new JButton("Remove Book");

JButton searchTitleBtn = new JButton("Search by Title");

JButton searchAuthorBtn = new JButton("Search by Author");

JButton issueReturnBtn = new JButton("Issue/Return Book");

JButton listBtn = new JButton("List All Books");

JButton saveBtn = new JButton("Save Books");


buttonPanel.add(removeBtn);

buttonPanel.add(searchTitleBtn);

buttonPanel.add(searchAuthorBtn);

buttonPanel.add(issueReturnBtn);

buttonPanel.add(listBtn);

buttonPanel.add(saveBtn);


// Text area for displaying the list of books

displayArea = new JTextArea();

displayArea.setEditable(false);

add(new JScrollPane(displayArea), BorderLayout.CENTER);
```

```

// Adding components to the window

add(inputPanel, BorderLayout.NORTH);

add(buttonPanel, BorderLayout.SOUTH);


// Button actions

addBtn.addActionListener(e -> addBook());

removeBtn.addActionListener(e -> removeBook());

searchTitleBtn.addActionListener(e -> searchByTitle());

searchAuthorBtn.addActionListener(e -> searchByAuthor());

issueReturnBtn.addActionListener(e -> issueReturnBook());

listBtn.addActionListener(e -> listBooks());

saveBtn.addActionListener(e -> saveBooks());


setVisible(true);
}


// Load books from file

private ArrayList<Book> loadBooks() {

    try (ObjectInputStream ois = new ObjectInputStream(new FileInputStream(FILE_NAME))) {

        return (ArrayList<Book>) ois.readObject();

    } catch (Exception e) {

        return new ArrayList<>();

    }

}


// Save books to file

private void saveBooks() {

```

```
try (ObjectOutputStream oos = new ObjectOutputStream(new FileOutputStream(FILE_NAME)))
{
    oos.writeObject(books);

    JOptionPane.showMessageDialog(this, "Books saved successfully.");
} catch (IOException e) {
    JOptionPane.showMessageDialog(this, "Failed to save books.");
}
}
```

// Add book to the library

```
private void addBook() {
    String title = titleField.getText().trim();
    String author = authorField.getText().trim();
    if (!title.isEmpty() && !author.isEmpty()) {
        books.add(new Book(title, author));
        titleField.setText("");
        authorField.setText("");
        JOptionPane.showMessageDialog(this, "Book added.");
    } else {
        JOptionPane.showMessageDialog(this, "Please enter both title and author.");
    }
}
```

// Remove a book from the library

```
private void removeBook() {
    String title = JOptionPane.showInputDialog(this, "Enter title to remove.");
    if (title != null && !title.trim().isEmpty()) {
        books.removeIf(book -> book.getTitle().equalsIgnoreCase(title.trim()));
    }
}
```

```
JOptionPane.showMessageDialog(this, "Book removed.");

}

}

// Search for a book by title

private void searchByTitle() {

    String title = JOptionPane.showInputDialog(this, "Enter title to search:");

    if (title != null && !title.trim().isEmpty()) {

        books.stream()

            .filter(book -> book.getTitle().equalsIgnoreCase(title.trim()))

            .forEach(book -> JOptionPane.showMessageDialog(this, book));

    }

}

// Search for a book by author

private void searchByAuthor() {

    String author = JOptionPane.showInputDialog(this, "Enter author to search:");

    if (author != null && !author.trim().isEmpty()) {

        books.stream()

            .filter(book -> book.getAuthor().equalsIgnoreCase(author.trim()))

            .forEach(book -> JOptionPane.showMessageDialog(this, book));

    }

}

// Issue or return a book

private void issueReturnBook() {

    String title = JOptionPane.showInputDialog(this, "Enter title to issue/return:");
```

```

if (title != null && !title.trim().isEmpty()) {

    books.stream()

        .filter(book -> book.getTitle().equalsIgnoreCase(title.trim()))

        .findFirst()

        .ifPresent(book -> {

            book.toggleIssue();

            JOptionPane.showMessageDialog(this, "Book status changed: " + (book.isIssued() ?
"Issued" : "Returned"));

        });

    }

}

```

// List all books in the library

```

private void listBooks() {

    if (books.isEmpty()) {

        displayArea.setText("Library is empty.");

    } else {

        StringBuilder sb = new StringBuilder();

        for (Book b : books) {

            sb.append(b).append("\n");

        }

        displayArea.setText(sb.toString());

    }

}

```

```

public static void main(String[] args) {

    new LibraryManagementSystemGUI();

}

```

